# Live Scatterplots

James R. Cordy
Queen's University, Canada
cordy@cs.queensu.ca

## ABSTRACT

Scatterplots have been used to help understand clone relationships in large scale systems since the earliest large system studies more than a decade ago. They often expose interesting patterns of cloning between subsystems and point to opportunities for further analysis. However, the remaining question when such patterns are seen is always, "but what is that?" Live scatterplots are aimed at providing an immediate, intuitive answer that can help the analyst to quickly identify and access subsystems and clones involved in a pattern simply by directly pointing at it in the scatterplot. Live scatterplots exploit the table, title and hyperlink tags of standard HTML to provide this ability in any standard browser, without the need for custom frameworks.

## Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*code clones*

## General Terms

Measurement, Experimentation

## Keywords

software clones, clone analysis, visualization

## 1. INTRODUCTION

Since the early days of clone detection, scatterplots, two-dimensional plots in which the axes represent the files or directories of the system or systems and the points represent the presence or absence of a clone relation between them, have been used to abstract and better understand the distribution and density of cloning in large scale systems. For example, in the original CCFinder paper [1], Kamiya et al. used a scatterplot to visualize the cloning between Linux, FreeBSD and NetBSD. Over the years scatterplots have been refined to add colour heatmaps [2] and other modifications to better indicate clone density, size or other characteristics, allowing for better abstraction and interpretation of large scale clone results.

However, a difficulty with scatterplots has always been their relationship to source. Even when the axes are reduced to only include those subsystems involved in cloning, the scale of points in the scatterplot is still so dense that it is difficult to see which subsystems are involved in interesting patterns, and what sources these patterns are attached to. If the axes are left unreduced, the plot is in general too dense even to see patterns effectively, and the relation to even high-level subsystems cannot be discerned from the plot.

As a result of these limitations, a lot of work has been done on other ways to visualize higher level cloning relationships, using tree diagrams, graph layouts, and other representations using sophisticated frameworks to assist in understanding. Recent such systems include those by Jiang and Hassan [3] and Göde and Koschke [4] for understanding clone evolution in large scale systems. In Live Scatterplots we explore a simpler idea: enhance the original scatterplots with interactive information available by simply pointing at the area of the plot that looks interesting. Gemini [6] provides a version of this idea, using click-through to clone pairs and source, but we have in mind something both simpler and more general, that is not tied to any particular clone detector, custom framework or level of abstraction.


*IWSC 2011* May 23, 2011, Waikiki, Hawaii, USA


## 2. SCATTERPLOTS AS WEB PAGES

Live Scatterplots is a tool that accepts the output of any clone detector (currently the XML output form of the NiCad clone detector [5]), and automatically renders the clone pairs as an interactive web page that displays clones as a scatterplot over the subsystems (directory structure) of the original source system or systems (Fig. 1). The axes are specified as two files of directory names to be used for the rows and columns of the plot respectively. These files specify the level of abstraction to be used. They can be source file names (for small systems), low level source directories (for mid-sized systems) or higher-level subsystem directories (for very large systems). The axes can use the same set of directory names (if the plot is of clones in a single system) or two different sets (for clones between two systems).

Implementation of scatterplots as web pages is very simple. The tool simply examines each clone pair reported by the clone detector, pattern matches the first fragment's file path with the directories of the rows and the second with the directories of the columns, and forms a matrix with the pairs in the corresponding cells. When all pairs have been placed in the matrix, Live Scatterplots renders the matrix as an HTML table, using cell colour to represent clone density, yielding a web page like the one shown in Figure 1.

Live Scatterplots can display either the entire matrix, or can reduce the axes to include only those directories which actually contain clones. In either case, rendering is fast and accurate in modern browsers. Live Scatterplots can render the table with either a black background, to more easily see low density cloning (Fig. 2), or a white background (Fig. 1), to better see very dense plots, and with or without labels.

## 3. LIVE SCATTERPLOTS

Besides the obvious advantages of portability and simplicity associated with browser-based rendering, the big advantage of HTML-based scatterplots is that it is easy to make them interactive. We exploit the capabilities of HTML to

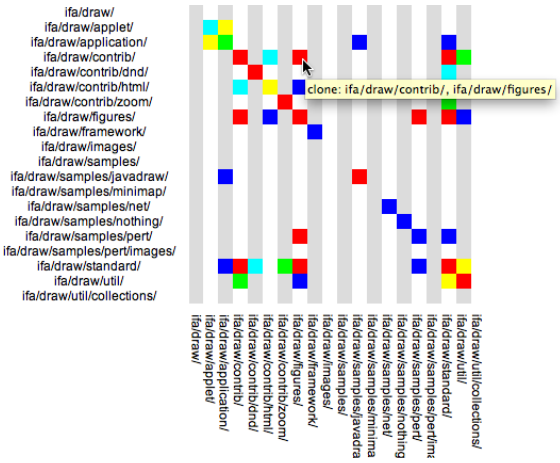


**Figure 1: Live Scatterplot of block clones in JHotDraw**

*Rendered in Safari. Hovering over a cell pops up the locations of the cloned fragments. Colour represents clone density.*

add interactive labelling and links to attach areas of the scatterplot directly to information such as the most important question we want to know - "what is that pattern?".

Interaction can be added in two ways - a "title" attribute for each table cell giving the source file and clone information for the cell when hovering over it, and a hyperlink to the corresponding source directories to follow when clicking it. If we are using NiCad, links can also lead to the corresponding clone classes in the clone report web pages.

The title attribute is however the main thing that makes Live Scatterplots "live". By simply hovering the mouse over any particular area in the scatterplot, the browser automatically displays the cell title, giving us immediate detailed information about the corresponding subsystems or clones. This information is exactly what we are looking for when we pose the question "what is that?", and immediately enhances our understanding of the patterns visible in the scatterplot.

## 4. DEMONSTRATION

Live Scatterplots are implemented as a small Turing program that takes as input the row and column directory names to be used for the axes and the clone pairs output by the clone detector (currently NiCad). It renders scatterplots as HTML web pages in seconds, even for very large systems such as Linux. In the demonstration we will run the tool on large systems and display the results in a standard browser on a laptop computer.

Figures 2 and 3 show examples of interactive web pages generated by Live Scatterplots for a cross-system clone analysis of FreeBSD and Linux. Figure 2 shows the overall function cloning between the systems using blind renaming and a 30% near-miss threshold. The popup tells us that the dense clone pattern we are pointing at is the ACPI driver code in the two systems. Figure 3 is the scatterplot for near-miss exact clones between the systems, abstracted to only those directories where cross-cloning is present. The popup tells us that the line of red on the lower right corresponds to exact cloning between the XFS subsystems of the two systems.

## 5. SUMMARY

Live Scatterplots is by no means a brilliant insight. On the contrary, it is a very simple practical idea that makes scatterplots easy, portable and more directly understand-

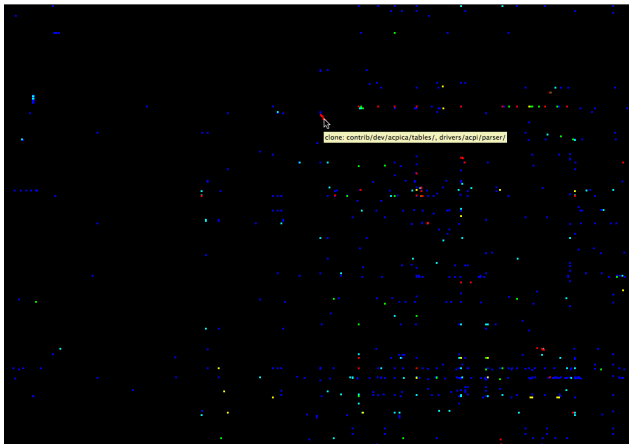


**Figure 2: Live Scatterplot of Linux (X axis) vs FreeBSD (Y axis) near-miss function clones**

*Colours reversed and enhanced to expose sparse clones.*

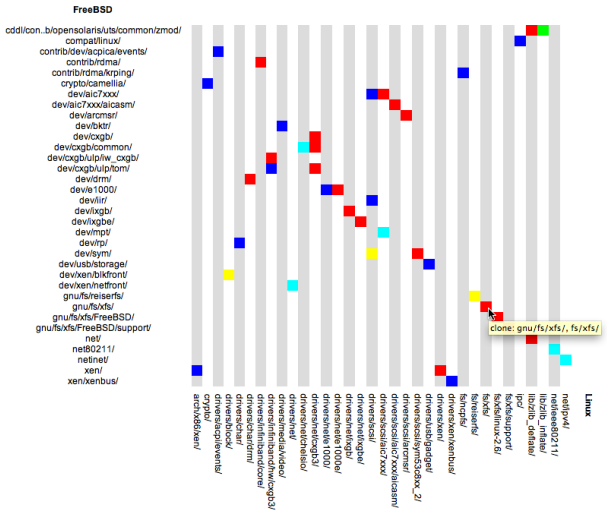


**Figure 3: Linux vs FreeBSD exact clones (axis-reduced)**

able without the need for custom visualization applications or frameworks. By exploiting the tabling and interaction capabilities of standard HTML browsers, it yields interactive scatterplot rendering that is simple, portable and efficient.

## 6. ACKNOWLEDGEMENTS

This work is supported by the Natural Sciences and Eng. Research Council of Canada (NSERC), and by an IBM CAS faculty award.

## 7. REFERENCES

[1] T. Kamiya, S. Kusumoto, K. Inoue. CCFinder: a multilinguistic token-based code clone detection system for large scale source code. *IEEE Trans. Softw. Eng.* 28(7):654-670, 2002.

[2] S. Livieri, Y. Higo, M. Matushita and K. Inoue. Very-large scale code clone analysis and visualization of open source programs using D-CCFinder. In *ICSE*, pages 106-115, 2007.

[3] Z.M. Jiang and A.E. Hassan A framework for studying clones In large software systems. In *SCAM*, pages 203-212, 2007.

[4] N. Göde and R. Koschke. Studying clone evolution using incremental clone detection. *J. Softw. Maint. and Evol. - Research and Practice* (2011, to appear).

[5] C.K. Roy and J.R. Cordy. NICAD: Accurate detection of near-miss intentional clones using flexible pretty-printing and code normalization. In *ICPC*, pages 172-181, 2008.

[6] Y. Ueda, T. Kamiya, S. Kusumoto, and K. Inoue. Gemini: Maintenance support environment based on code clone analysis. In *METRICS*, pages 67-76, 2002.